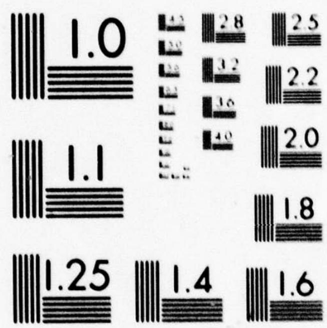
AD-A072 245 SYSTEM DEVELOPMENT CORP MCLEAN VA F/G 9/2
COUNTERMEASURES.(U)
JUN 79 N00173-78-C-0455
UNCLASSIFIED SDC-TM-WD-7999/000/02 NL

1 OF 1
AD
A072245

END
DATE
FILMED
9-79
DDC



3

System Development Corporation

LEVEL II

TM-WD-7999/000/02
FINAL

COUNTERMEASURES

JUNE, 1979

System Development Corporation

14 SDC-TM-WD-7999/000/02

9 FINAL rept.

6 COUNTERMEASURES.

12 75p.

11 JUN[redacted]979

15

CONTRACT NO.: N00173-78-C-0455

79 08 1 028

THIS DOCUMENT HAS NOT BEEN CLEARED FOR OPEN PUBLICATION.

339 860

JLB

TABLE OF CONTENTS

APPENDIX _

COUNTERMEASURES

1.1 INTRODUCTION

This appendix describes countermeasures that will reduce the vulnerability of an ADP facility. The countermeasures described herein are a representative group for improving overall computer security. They are to be used to assist ADP installations in performing a risk assessment. The format for each of the proposed countermeasures is as follows:

a. Vulnerability. The statement of a security vulnerability in an ADP facility.

b. Countermeasure. A brief description of a security countermeasure that can reduce the stated vulnerability.

c. Confidence. The level of confidence, e.g., low, medium, high, that can be placed in the proposed countermeasure.

d. Cost. A qualitative statement of the cost or (if possible) dollar costs that would be incurred by implementing the proposed countermeasure. Cost figures are estimates based on a cost of $60,000 per man-year.

e. Caveats. The limitations, unusual risks, dependencies, and/or disadvantages related to the countermeasure.

Section 2.1 of this appendix contains references (applicable policy and/or technical reference) pertaining to each countermeasure.

1.2 SOFTWARE

1.2.1 Security Audit Trails

Vulnerability. Deficient protection features for the operating system may allow actual or attempted security violations to go undetected without an adequate audit trail capability.

Countermeasure. Establish an audit trail capability and ADP system security officer (ADPSSO) review process. The audit trail should provide accurate information on security-related transactions. From the data contained in the audit trail, the ADPSSO should be able to answer the following questions:

a. Who attempted to log onto the system (name, user ID, password, time of of day, terminal identification [if applicable])?
b. Who was on the system (name, user ID, password, time of day, terminal identification [if applicable])?
c. What files were requested?
d. What was the nature of the access -- read, write, execute, append, or delete?
e. What files were created?
f. Was any output produced?
g. What date and time did a transaction occur?

Once an appropriate data base has been built, it is possible for the ADPSSO to check the current audit trail against historical use patterns and identify security-related exceptional usage patterns. Note that it may be difficult to identify security violations as such and that an audit trail may not be feasible in some systems.

Confidence. An average security audit package is rated low to medium.

Cost. If the software currently in use supports audit trail information gathering and journalizing, the main cost of establishing an audit trail will be in the area of computer system overhead. If audit trail information-gathering and journalizing software has to be written, the cost associated with this countermeasure may range from two to four man-years of effort ($120,000 to $240,000). In either case once software is implemented to create an audit trail, there will be a need for additional personnel time for analyzing the data. A large system may need one person at half-time ($30,000).

The costs associated with the software development to support this countermeasure would probably be more than any one site would consider reasonable. To bring down costs for any one site, costs of software development can be shared by several sites with similar hardware and system software configurations.

Caveats. An audit trail can provide a deterrent effect, but that effect can be lost if it becomes generally known that the audit trail is not subject to scrutiny.

An audit trail provides little defense against software penetration.

The level of confidence for auditing depends on the protection afforded to the audit software. If the software can be easily disabled, the confidence is lower.

1.2.2 Threat Monitoring.

Vulnerability. Inadequate protection features for an operating system may invite unauthorized access to the system, misuse of resources, or other undesirable activities.

Countermeasure. Threat monitoring is a preventive measure that can recognize an attack and quickly notify an appropriate authority. Notification may be by an alarm on the operator's console, a message to the operator or ADPSSO (if on-line), or an automatic dial-up of the security officer's phone number.

Threat monitoring (or a surveillance program) is installed as software that has access to appropriate security information about the users, files, and processes of the system. The software constantly monitors the activities of the system and attempts to recognize unusual activities or patterns.

When threat monitoring is advertised as an operating system feature, it can serve as a useful deterrent to unauthorized activities.

Confidence. This countermeasure is given a medium level of confidence.

Cost. A surveillance program of this type requires from 12 to 24 months of development effort ($60,000 to $120,000). A hardware feature to sound an alarm or dial a security officer when a violation takes place may cost up to an additional $5,000.

Caveats. A threat monitoring program can be expected to degrade the performance of an operating system by interrupting normal processing to monitor security. The extent of the degradation depends upon the number of activities monitored, the frequency of monitoring, and the resources used to evaluate the legitimacy of system activities.

A threat monitor will not provide substantial defense against software penetration.

1.2.3 Residue Control

Vulnerability. An operating system may allow sensitive information to remain in public primary and secondary storage. This information may be compromised by browsing[1] attacks.

Countermeasure. Purge or erase all publicly accessible storage areas before allowing a program to use them. A software program can erase storage areas, e.g., sort work areas, temporary data files, input/output buffers. Some computers may provide a hardware clear switch for manually clearing memory.

Confidence. A high level of confidence can be placed in this countermeasure.

Cost. Software development costs can range from one month to several years of effort ($5,000 to $300,000). The Clear Memory Utility (CMU) project for the Honeywell 6000 series required approximately 6 man-years. CMU cleared main memory, control processor registers, and the microprogrammable controllers

---

[1] Browsing is defined as searching through storage to locate or acquire information without necessarily knowing of the existence or the format of the information being sought.

between periods of secure processing for the World Wide Military Command and Control Systems (WWMCCS).

Caveats. Associated with this countermeasure will be the overhead to execute the software.

This countermeasure is not to be confused with the overwrite of storage or magnetic media prior to its physical release to an uncleared facility. Because of the hysteretic properties of magnetic memory and storage media, a single overwrite in that case is not sufficient. Additional measures must be used to prevent previously stored data from being recovered under laboratory conditions.

1.2.4 Log-on Attempts

Vulnerability. Systems may be deficient by permitting an unlimited number of log-on attempts. An unauthorized user who is trying to log-on by guessing the log-on procedure may go unnoticed by the system. If the unauthorized user does guess the log-on procedure or password, the system becomes susceptible to compromise.

Countermeasure. Several approaches can be associated with the number of allowable unsuccessful attempts to log-on to the computer system. They include the following alternatives:

a. Permitting an unlimited number of attempts.

b. Allowing one attempt and then automatically locking the terminal out of the system.

c. Specifying a fixed number of attempts and then automatically locking the terminal out of the system.

d. Surveillance of the terminal session after several failed attempts.

The decision of how many attempts to allow a user to log-on to a computer system is a policy decision for the host system's ADPSSO to enforce.

Confidence. Alternative a provides no confidence. A very high level of confidence is gained by b or d. A high level of confidence can be placed in c if no more than two or three attempts are allowed.

Cost. The cost associated with this countermeasure depends on whether the current software supports the counting of attempted log-ons. If the necessary software is in place and only the number of unsuccessful attempts has to be changed, the cost may be less than $100. If the software must be developed, the cost of developing or purchasing the required software should be less than $5,000. Sharing the cost over several installations with comparable software and requirements can reduce the cost per installation.

Caveats. The number of log-on attempts to be permitted will vary depending on such things as the following:

a. The trustworthiness of users.
b. How closely the terminal areas are monitored.
c. The sensitivity of the data contained in the system.
d. Whether or not dial-up access is provided.

1.2.5 Remove Vendor-Supplied Passwords.

Vulnerability. Most vendor-supplied software comes with standardized and well-known imbedded passwords. Vendor-supplied passwords are often provided with operating system software, data management and file systems, and various utility packages. These standardized passwords serve to facilitate the installation of such software. A large community of users knows these passwords. If they are not changed the system can be penetrated easily.

Countermeasure. Change the imbedded passwords that come with the vendor software. Provide unique passwords for each site and protect them to the highest level and most restrictive category of information processed by the ADP

system. Passwords should be randomly generated and distributed by the ADPSSO or a delegated member of the officer's staff.

Confidence. Randomly generated passwords rate a high level of confidence. User-selected passwords are rated from low to medium.

Cost. The cost of generating random passwords to replace the vendor-supplied passwords is less than $500.

Caveats. A mathematically sound random number generator and a well-administered distribution scheme can be negated by careless employee practices, such as failing to safeguard passwords. In some cases, randomly generated passwords are unpronounceable or hard to remember. Such passwords tend to get written down, thus becoming subject to compromise.

1.2.6 Password Protection from Visual Observation.

Vulnerability. If the password of an authorized user of a system is displayed on a terminal screen, on the terminal hardcopy, or on the batch hardcopy, the password may be compromised.

Systems may be deficient by not providing this basic protection.

Countermeasure. Provide a mechanism to protect passwords from being displayed on the terminal screen or hardcopy. Provide software that will either suppress printing of the password when entered, or present a strikeover field onto which the terminal operator can enter a password.

In the case of password protection for batch jobs, modify the job control language (JCL) handlers so that the password is removed before the JCL cards are printed as part of the execution report. Card decks containing passwords must have procedural safeguards.

Confidence. Suppressing the printing of the password gains a high level of confidence. Other protection mechanisms are rated medium.

Cost. The cost for developing software to suppress password printing requires one to two man-months ($5,000 to $10,000).

Caveats. This countermeasure will not prevent employees from sharing passwords or from writing the passwords on desk pads, calendars, or from discarding card decks containing passwords.

1.2.7 File Encryption.

Vulnerability. Operating system flaws may permit unauthorized access to sensitive files. Inadequate protection of off-line magnetic media such as tapes and disks may result in an unauthorized disclosure.

Countermeasure. Sensitive data files can be encrypted to reduce the possibility of compromise through disclosure. Cryptographic schemes can also provide an indication that files have been modified. This countermeasure provides protection while files are in an encrypted state. However, while files are being processed as cleartext (unencrypted), this countermeasure provides no protection. The possibility of compromise is reduced since the information in the file appears as cleartext only while being processed. The National Bureau of Standards has approved a Data Encryption Standard (DES) for protecting certain types of sensitive information.

Confidence. This countermeasure provides very high confidence that encrypted files will not be disclosed due to loss of an off-line medium. Confidence against disclosure due to operating system flaws is medium.

Cost. The use of the DES in software is not approved at this time. Approved hardware implementations are available from the Collins Group of Rockwell International, IBM, Motorola, Intel, Burroughs, and Fairchild. Contact these vendors to obtain cost figures.

Caveats. An approved encryption device must be used in conjunction with special administrative and key management procedures to provide secure operation. The encryption keys must be protected at all times.

1.2.8 Data Base Protection.

Vulnerability. Without proper data base protection measures, information contained within a data base may be compromised. Data bases may be compromised by asking a set of queries which return only statistical information and making inferences about a specific entry from the results of the set of queries.

Countermeasure. Several methods may be applied to reduce the possibility of compromise of a data base.

a. Inoculate the data with random errors to make the data base less precise. The majority of the data must be inoculated so that individual records are not necessarily accurate, but that the overall data base is still accurate for statistical analyses.

b. Use threat monitoring and logging to detect attempts to compromise data. For example, check unusual overlap patterns created by successive queries.

c. Identify the public characteristics of an individual item on the data base. Permitting queries on a subset of these characteristics will prevent an unauthorized user from determining the individual item. However, it is possible to deduce a small range of values for the individual item and then use outside information to determine the exact value (see reference a).

d. Use link files to separate identifying characteristics of data items from the statistical data associated with each item. One file contains the identifying characteristics of a data base item. Another file contains statistical data associated with the data base item. A link file matches the contents of these two files. Separating the data base in this way makes it more difficult to associate specific data items with identifying information.

e. Restrict the types of queries that can be made against the data base. For example, do not allow queries against certain combinations of data items.

Restrict queries against small subsets of the data base. Certain subsets of the data base may have strong individual characteristics.

Confidence. Countermeasure a is rated medium. Countermeasures b and d are rated low. Countermeasures c and e are rated from medium to high.

Cost. The costs for developing b can range from one to two man-years of effort ($60,000 to $120,000). Costs for a and c are primarily in analysis work. An expected range may be from one to four man-weeks ($1,250 to $5,000). Costs for countermeasure d are development costs and range from three to six man-months ($15,000 to $30,000).

Caveats. Techniques to compromise data bases are quite sophisticated and are far more advanced than most countermeasures that can be readily implemented to protect data bases. This type of countermeasure can serve to deter unsophisticated attempts to compromise a data base.

1.2.9 Periodic Inspections of Software.

Vulnerability. Software may have intentionally placed trojan horses, trap doors, or similar modifications that can cause unauthorized disclosure, unauthorized modification, destruction of data, or denial of service.

Countermeasure. Conduct periodic inspections of software in several ways such as the following:

a. Make visual inspections of program listings and files to detect unusual instances of data or software.

b. Perform automated code matches. A program can be developed to compare files for exact matches. These files can contain software or data.

c. Verify the date that a file was last modified. Compare this date against the date when the file was last modified for authorized purposes. This countermeasure requires that the system is able to maintain the last date of access to a file.

d. Compute and securely store checksums of software and data files. Then periodically checksum each file and compare the result to the stored checksum. A checksum is computed based on a portion of the data in each record.

Confidence. Countermeasures a, b, and d are rated medium. Countermeasure c is rated low.

Cost. Costs for countermeasures a and c are primarily personnel time. For each inspection the cost should average $600 (2.5 man-days). Costs for countermeasures b and d are primarily software development and the machine time to perform the inspection. Countermeasure b should cost about $5,000 (one man-month of development). Countermeasure d should cost only about $1,250 (one man-week of development effort).

Caveats. Confidence in these countermeasures depends on the secure storage of original software, data, checksums, and lists. Confidence and cost depend on the detail and frequency of the inspections.

1.2.10 Controlling Use of Assembler Language Coding.

Vulnerability. Software may be developed to penetrate the operating system. Assembly language provides the most direct access to hardware and software features that may be manipulated to penetrate the operating system.

Countermeasure. The following alternatives may be used to minimize the vulnerability to operating system penetration by means of assembly programs:

a. Remove the assembler language processor from the ADP system.

b. Control access to the assembler language processor through the use of passwords (limit the issuance of these passwords to those programers, e.g., system programers, who have a valid requirement to use assembler language).

c. Place the assembler language processor on an off-line storage medium so that it cannot be used without the active cooperation of the computer console operators who will have to mount the off-line storage medium.

Confidence. Countermeasure a is rated very high, b is medium, and c is high.

Cost. Each of the above countermeasures should cost less than $1,000 to implement. A possible exception is b, if the system in use does not already support password protection. In this case, the cost can be expected to exceed $10,000 for the cost of procuring new system software.

Caveats. Most applications programs can be written in a higher order language. Some application programs must use assembly language. Examples include real-time programs, terminal handlers, or data base manipulation programs.

This countermeasure does not address the problem of using a higher order language to create executable code and then transferring control to that code.

1.2.11 Two-Person Control

Vulnerability. Deficient security procedures may permit unauthorized modifications to be made to system software that controls log-on procedures, password verification and replacement, audit trail journalizing, and storage purging. Unauthorized modifications are more easily accomplished if the update procedure can be accomplished by a single individual.

Countermeasure. Require more than one person to make modifications to system software that controls log-on procedures, password verification and replacement, audit trail journalizing, and storage clearance. A second qualified individual should authorize or supervise modifications that are being made.

Confidence. With two-person control of system software, a medium to high level of confidence can be assured that unauthorized modifications are not being made.

Cost. The costs for this countermeasure will be for the increased personnel requirements.

Caveats. To the extent possible, this countermeasure should also be applied to application programs.

1.2.12 Periods Processing.

Vulnerability. Most general-purpose operating systems do not provide adequate controls to keep users from gaining unauthorized access to data.

Countermeasure. When several levels of classified or sensitive information must be processed, consider implementing periods processing. Periods processing is defined as a period of time during which information of a given security level is processed. Each classification level is processed at different times and the system is purged between periods. This requires a well-conceived and carefully followed checklist for shifting from period to period (see 1.3.3, Hardware Configuration Control). Periods processing requires procedural controls to insure that all users are cleared for the highest classification and most restrictive category of information being processed during the period.

Confidence. A very high level of confidence can be gained that an unauthorized access will not occur because of improper classification level.

Cost. The costs of this countermeasure include the administrative task of developing the necessary procedures for implementing periods processing and lost computer time while shifting from one period to another.

Caveats. There are several disadvantages in periods processing:

a. The computer is not available during the switch-over between periods. This may represent a significant overhead cost in terms of lost processing time.

b. Separate versions of the operating system software, with unique classification level requirements, must be used and maintained for each period.

c. The ADP system will be available to individual users only during their authorized periods.

d. The turnaround time for particular jobs may be very lengthy.

e. The procedures to purge the system when changing periods may be extensive, costly, and prone to error.

1.2.13 Testing and Debugging.

Vulnerability. The procedures for testing and debugging software must may be inadequate. If there is a software failure during program testing or debugging, it may be difficult to ascertain the state of the computer and insure the integrity of data that was on-line or otherwise readily accessible. In the period of system instability during a software failure, normal system safeguards may not be in effect. Data may be disclosed inadvertently, e.g., misrouted to an unauthorized user.

Countermeasure. Two sets of countermeasures can be employed, one set for systems programs, the other for application programs.

a. System programs. The testing and debugging of system software programs should be performed initially during dedicated time in a controlled environment. If operational user files are required for testing, copies of these files should be used. Operational testing may be carried out when quality assurance personnel are satisfied that the programs are operating reliably.

b. Application programs. The testing and debugging of applications programs may be permitted during nondedicated times, but only copies of data files should be used.

Confidence. The proposed countermeasures insure a medium level of confidence that users will not be seriously interrupted and that data contamination will not occur during program testing and debugging.

Cost. The costs associated with these countermeasures are variable. They are administrative in nature; that is, the separation of production and debugging time must be enforced. Also, some time is lost to users when the system is dedicated to the testing and debugging of system programs.

1.2.14 Security Editing and Accounting.

Vulnerability. Deficient input/output procedures may damage the integrity of operational files. As a result, incorrect decisions may be made based on the invalid data.

Countermeasure. Use strong edit and transaction accounting features to insure data integrity. Some of these features are the following:

a. Control on input such as transaction counts, batch totals, card verifier operations separate from keypunching, self-checking number device on keypunch, and machine-readable document input. Types of input validiation checks include: character checks, such as testing for numeric, alphabetic, or specific character group, blanks, field separators or special characters and the proper or valid arithmetic sign; field checks such as testing for limits, ranges, valid item, consistency, sequence, etc.

b. Controls on processing such as transaction counts, batch control totals, hash totals for batch, validation by file reference (does a record exist for this item?), consistency checks (does this item agree with previously stored data?), control on rounding errors, etc.

c. Control on ouptut such as item counts, control totals, trailer labels on data sets, control records, serial numbers on documents, e.g., checks or invoices.

Labels on tapes or discs may contain label identifier, file number, batch number, creation date, retention cycle (son, father, grandfather), volume number, e.g., reel number, a count of the records on the file.

Examples of an input/output control group's typical responsiblities include the following:

(1) Log in jobs received for processing from user departments.

(2) Check document counts and control totals of work received.

(3) Notify user department that the work has been received and indicate whether the counts and totals are correct.

(4) Note any work that was due but not received.

(5) Note and initiate action on any improper preparation by the user departments, such as failure to provide counts or totals.

(6) Submit documents to be keypunched or entered onto tape or disk.

Confidence. Strong edit and accounting features cannot totally prevent the subtle alteration or corruption of data. However, a high level of confidence can be assured that well-conceived procedures can detect most input and output data errors.

Cost. Costs will be primarily for development and programming. Depending upon the level of detail for the integrity controls, the costs can range from two man-weeks ($2,500) to six man-months ($30,000).

Caveats. Edit and accounting features will degrade system performance slightly by requiring added processing of input/output.

1.2.15 Software Engineering Tools.

Vulnerability. The failure of software to perform according to requirements has the potential to compromise security. Software failure may, for example, destroy the integrity of data bases or allow inventory shortages to go unnoticed.

Countermeasure. Below is a representative sampling of the many software tools available. These tools aid the development process to provide increased confidence that software will perform reliably and in accord with requirements.

a. RISOS (Research In Secure Operating Systems, project at Lawrence Livermore Labs) tools were developed to analyze assembly language programs.

Analytical tools available in RISOS include a program that counts occurrences of a specified symbol, a program that identifies the control flow and flags specified items, and a program that locates instruction patterns. These are some of the very few software engineering tools developed specifically for security.

b. Software quality measures are computer programs that examine a program to generate a quantifiable measure of the program's quality. This allows testers to reject programs with quality measures that are outside a certain range, on the assumption that program reliability decreases as quality decreases.

c. Self-metric software examines the source code of a computer program and inserts software measurement probes. The software probes help testers estimate the extent to which a program has been tested by some set of test data. Data gathered from such probes might indicate the number of times a loop was executed, the entry and exit values, and the test stimuli provided.

d. Test data generators are computer programs that generate test cases to be used in the testing of software. These programs range from utility type programs that generate sequences of alphanumeric and/or numeric data based upon parametric inputs, to entire systems that interpretively examine the flow through a program and attempt to generate appropriate sequences of test cases.

e. Audit programs insure that programs conform to a given set of programing standards. Programs that deviate significantly may be more difficult to understand and may have flaws that could affect security.

f. Trace programs record data such as program variables or events that can assist in program debugging and validiation.

Confidence. Confidence placed in these tools ranges from low to high.

Cost. Tools that are readily available, such as those developed for the U.S. Government, may be obtained at no cost. Other tools may be purchased or

leased. Contact the supplier of these tools for cost information. Development cost of tools can range from one to four years ($60,000 to $240,000).

Implementation costs vary for each tool and range from one man-day to two man-weeks of effort ($250 to $2,500), depending on system compatability.

Application of the tool varies depending upon the effort made in developing test cases, standards or criteria. Costs may range from one man-day to a man-week of effort ($250 to $1,250).

Caveats. Different software engineering tools accomplish different goals. The confidence placed in each tool is based upon the expected improvements in reliability and conformance with requirements as a result of using the tool.

1.2.16 Secure Subsystems.

Vulnerability. Most general-purpose operating systems are unable to enforce security policies without stringent administrative and procedural controls. Generally, it is impractical to retrofit security into existing operating systems by attempting to correct all known flaws.

Countermeasure. A secure subsystem approach may provide an adequate level of security if most of the users of a computer system are application program (subsystem) users and have no need for a general programming capability. Secure subsystems divide users who are concurrently active in a computer into isolated groups that support distinct operational missions. User group isolation restricts access to security-related objects based upon the different need of each user to know the information contained within the objects. Such differentiation can be important even if the users who are to be isolated have identical security clearances. In certain benign environments, secure subsystems can also justify a limited form of multilevel operation. A single-level secure subsystem could be certified to operate at a level lower than the system-high level to support a set of users cleared only to the level of the secure subsystem.

These subsystems would prevent a user from escaping from the subsystem. Moreover, the subsystem would provide security controls that may be lacking in the operating system.

Confidence. Secure subsystems assure a high level of confidence that the users of the subsystem are protected adequately from each other.

Cost. The costs associated with this countermeasure are unknown but can be assumed to he high. Costs will vary based upon the number of subsystems to be secured. The cost can be expected to be not less than $500,000. The cost may be shared by several installations having comparable software requirements.

Caveats. Secure subsystems protect users from each other but they do not guarantee protection from penetrators outside the subsystem.

1.2.17 Security Kernel.

Vulnerability. Most operating systems have weak security features. Programs may subvert the operating system to gain unauthorized access to data or a faulty operating system may malfunction, such as by misrouting data.

Countermeasure. For highly sensitive systems, consider employing a software security kernel. A security kernel is designed to mediate all access within the system and can generally be defined as the security policy enforcing code. Some characteristics of a kernel are that it is always invoked, it is tamper proof, and it is small. A small kernel is desired because formal specification and verification techniques are applied to prove consistency of successive levels of the design. It is extremmely difficult to verify formally a large body of specifications or code.

Confidence. The level of confidence to be placed in a security kernel is very high.

Cost. The costs to design, implement, and verify a security kernel are probably beyond the resources of any one organization. Costs may approach several million dollars. However, if the costs can be shared by enough

organizations with similar hardware and software the cost per organization can be reduced substantially. The KSOS effort is to provide an off-the-shelf security kernel.

Caveats. The security kernel is still a relatively new concept and there are few in existence. There are also unresolved questions concerning the effects of a kernel on system performance, the cost of formal verification, and the ability to maintain the software.

1.2.18 Virtual Machine Monitors (VMM)

Vulnerability. Many operating systems do not provide the level of protection required for certain applications.

Countermeasure. A virtual machine monitor (VMM) can isolate users from each other and offer a level of protection that most operating systems cannot. The VMM offers each user of the system its own virtual machine. Each virtual machine is provided by the VMM with a virtual CPU, virtual memory, virtual input/output channels, virtual devices, and virtual unit record equipment. A VMM does the following:

a. Interprets and executes privileged instructions.
b. Verifies input/output addressing and simulates the input/output devices.
c. Allocates hardware resources.

VMM's are commercially available from various vendors. The security of the systems is generally better than most operating systems. A secure VMM (reference e) is under development and should be comparable to a security kernel.

Confidence. The level of confidence is high that a VMM isolates users from each other.

Cost. A VMM that must be retrofitted into a current system would cost at least $500,000.

Caveats. VMM's would not be feasible for systems that require frequent interaction among programs. Also, the system overhead for the VMM is significant.

1.2.19 Password File Encryption.

Vulnerability. The file access control mechanisms in most general-purpose operating systems may not prevent a skilled penetrator from obtaining the on-line password file. This may lead to a penetration of the computer system and the unauthorized disclosure of information.

Countermeasure. The file containing the passwords used to log on to the system can be encrypted. Such a scheme will prevent an on-line password file from being readily intelligible if the file is disclosed. The password file is stored in encrypted form using a one-way or irreversible algorithm. The encrypted passwords can not be inverted to obtain the original cleartext passwords. In operation, user-supplied passwords are encrypted and compared against the encrypted passwords. A match indicates that a valid password was supplied. Presumably, if a penetrator is able to gain access to this file, then the other access control authentication mechanisms could also be bypassed. Encrypting the password file is an effective countermeasure against accidental disclosure and casual browsing.

Confidence. A high level of confidence can be placed in this countermeasure to protect against accidental disclosure and casual browsing. The confidence is rated medium for skilled penetrators.

Cost. Depending on the encryption scheme implemented, the cost can be expected to range from $5,000 to $50,000.

Caveats. When using such a scheme, recovering a forgotten password requires that a list of the passwords in unencrypted form be maintained manually.

There is also the danger that two or more clear passwords may produce the same "one-way transform." This may allow potential unauthorized access to

the system. One-way or irreversible algorithms have been broken in the past. Very short or user-selected passwords increase the possibility of compromise.

1.3 HARDWARE

1.3.1 Protection-State Variables.

Vulnerability. If a processor does not employ two or more protection-state variables, both the user and the operating system must operate in the same state. As a result, a user may be able to perform all hardware functions without restriction.

Countermeasure. A processor should have at least two protection-state variables, i.e., privileged mode/user mode, in which certain instructions are illegal except in privileged mode. Examples of privileged instructions include input/output, memory management, and context switching. Modification of the protection-state variables should be constrained by the operating system and hardware so that a program in user mode cannot switch itself into privileged mode.

Confidence. Depending upon how well the system software uses protection-state variables, this countermeasure ranges from low to high confidence.

Cost. The cost of this countermeasure is included in modern CPU costs.

Caveats. Procuring new hardware to recognize only two or more protection-state variables is rarely justified. New procurements should mandate hardware that meets this requirement and software that fully supports and exploits it.

1.3.2 Memory Protection Mechanisms.

Computer architectures may not have mechanisms to restrict main memory access by user programs. Lack of memory protection mechanisms also makes it possible for user programs to interfere either inadvertently or maliciously with other users or with the operating system itself.

Countermeasure. All computer system hardware that processes classified or sensitive information should support the use of memory protection mechanisms.

These mechanisms are designed to isolate users from each other and from the operating system. The hardware checks each fetch and store instruction for proper access.

Examples of hardware protection mechanisms include memory bounds registers (CDC 6000 Series), storage locks and keys (IBM 370 series), segmentation (IBM 360/67), paging (Honeywell 6180), rings, capabilities, tagged architecture (Burroughs B6700), and descriptor-based protection (Plessey 250).

Confidence. A computer architecture with a mechanism to restrict memory access rates a medium level of confidence. Architectures implementing segmentation, paging, rings, or other more restrictive mechanisms rate high or very high.

Cost. The cost of this countermeasure is included in modern CPU costs.

Caveats. New hardware procurements should include appropriate memory protection mechanisms.

1.3.3 Hardware Configuration Control (Periods Processing).

Vulnerability. Poor security procedures may make it possible for the system to be configured incorrectly following periods processing. This could lead to the unintentional storing of classified data on unclassified devices or the sending of classified data to a remote terminal that should have been disconnected.

Countermeasure. Establish and enforce the use of a configuration control checklist. This checklist should contain detailed procedures for connecting the individual ADP system components together into the specific system configuration to be employed during each period. These procedures include setting all hardware switches, powering up and down of each device, loading the standard software and firmware for the configuration system, system operating procedures, and shutdown and restart procedures. Strict adherence to the established procedures is essential for overall system security. To

insure that the procedures are followed, it is desirable that two people verify the new configuration.

Confidence. The strict use of a configuration checklist can provide a high level of assurance that the system is correctly configured for each mode of operation, such as top secret or unclassified.

Cost. The cost of developing a configuration control checklist is principally administrative. The cost of following this checklist is the time for the console operator and another person to verify the actual configuration against the checklist.

Caveats. This countermeasure is meant to be used when changing from one period to another during periods processing. It can also be used during start-up after the computer has been shut down.

1.3.4 Front-End Machines.

Vulnerability. Security functions such as password authentication, access control, and security monitoring may be rendered ineffective by penetration of the operating system.

Countermeasure. In some applications it may be desirable to employ a non-programmable minicomputer to perform functions such as password authentication, access control, and security monitoring for a larger co-located host.

Some advantages of this device are the following:

a. Off loading security functions may improve the performance of the host computer.

b. Locating security functions in a physically separate computer reduces the possibility that these functions may be compromised due to host computer penetration.

c. Isolating the security functions may allow for more rigorous software verification of security functions than is feasible in a host computer system.

Confidence. There are many variables to consider when assessing the level of confidence for this countermeasure. These include the strategy for attaching the front-end machine to the host, the overall design of the front-end software, and the techniques used to verify the design and implementation. However, if the front-end minicomputer is non-programmable, a high level of confidence is assured that the security functions will not be subverted.

Cost. The costs associated with a front-end machine can be expected to be not less than $250,000.

Caveats. A detailed cost benefit analysis is probably necessary to weigh the substantial development cost against the expected benefit.

1.3.5 Data Base Machines

Vulnerability. A penetrator of an operating system could have virtually unrestricted access to all information available on the on-line storage media managed by the operating system.

Countermeasure. A data base machine can be used in applications employing very large data bases or data bases that are shared by different computers, local or remote. The data base machine is a minicomputer between the main computers and the on-line storage media to manage the reading and writing of the data base. The authority of a user to gain access to the system can be checked in the data base machine. Other security-related considerations are:

a. The data base machine's operating system and hardware provide protection to the Data Base Management System (DBMS) and to security-relevant data.

b. There is a forced invocation of the DBMS to access data. Any attempted access of the data base causes a hardware interrupt that activates the DBMS.

c. The DBMS is separated from all user programming and direct access capabilities.

d. Hosts may continue to operate system-high and untrusted.

Confidence. Use of a data base machine would assure a medium level of confidence that a penetration of the host operating system would not allow unrestricted access to on-line data.

Cost. The hardware and software required for putting a data base machine into operation may cost $500,000 or more if developed from scratch.

Caveats. Some processing power would be gained in the host computer by transferring the data base management software to the data base machine. This may be an alternative to expanding the capacity of a saturated host computer. This performance gain could be outweighed by the overhead associated in communicating with the data base machine.

1.3.6 Redundant Equipment.

Vulnerability. In some situations even short periods of downtime due to equipment failure may pose a serious threat of denial of service if there is no backup hardware or contingency plan.

Countermeasure. In systems with a highly critical uptime requirement, install enough redundant equipment to carry on the minimum critical functions in the event of an equipment failure in the main configuration.

Confidence. Installation of sufficient redundant equipment will assure a high level of confidence that a denial of service will not occur due to equipment failure.

Cost. The costs associated with this countermeasure may be high depending on how much hardware must be duplicated.

Caveats. Rarely will it be necessary to provide redundancy for the entire hardware configuration. It should be necessary to duplicate only the minimum configuration of hardware to process the functions for which the agency or department cannot suffer a denial of service.

1.3.7 Hardware Error and Tampering Detection.

Vulnerability. Undetected hardware errors or hardware tampering may compromise security.

Countermeasure. Provide hardware with facilities to detect and expose internal hardware malfunctions. Modern hardware normally has error detection capabilities, such as parity error detection. Hardware components should cause an interrupt to occur whenever there is a change in their status. Software can then be developed to interpret the interrupt for possible tampering or change in hardware configuration. Software may also be developed to detect unusual error or interrupt patterns.

Confidence. Error and detection mechanisms assure a medium level of confidence that tampering and errors will not disrupt system functions.

Cost. Costs will be primarily in developing software to support these special hardware features. Costs may range from one man-month to one man-year of effort ($5,000 to $60,000).

Caveats. In addition to detecting tampering and errors, it would be desirable to implement recovery techniques for these problems. Costs vary according to sophistication of software detection capabilities.

1.3.8 Interruption-Resistant Power.

Vulnerability. The power supply for the ADP system may be inadequate to meet the facility's performance requirements.

Countermeasure. To correct for minor power line fluctuations (transients), install a voltage regulator transformer. This regulator will provide protection against minor transients and brownouts.

Confidence. High.

Cost. Approximately $100 to $200 per kilovolt-ampere (KVA) of load.

Countermeasure. Protect against short-term power failures by using a motor alternator with an energy storage flywheel. This configuration will provide up to 15 seconds of power and also protect against transients and brownouts.

Confidence. High.

Cost. Approximately $200 to $300 per KVA of load, plus cost of installation.

Caveats. A special room may be needed for the equipment.

Countermeasure. Protect against long-term power failures by using batteries. Depending on the ampere-hour capacity of the batteries and the KVA requirements of the ADP equipment, the load may be supported for up to two hours.

Confidence. High.

Cost. Approximately $700 to $900 per KVA of load, plus cost of site preparation and installation.

Countermeasure. To prevent a major loss of power, install a backup generating system.

Confidence. High.

Cost. Approximately $100 per KVA of load, plus installation and site preparation costs.

Caveats. In ADP facilities where environmental controls must be maintained for continued operation of the ADP facility, the additional electrical load has to be added to the generating capacity. Other electrical requirements that may be necessary for operation should be considered, such as lighting, alarm systems, and security systems.

1.4 PROCEDURES

1.4.1 Software Development Procedures.

Vulnerability. Software development procedures at the ADP facility may be inadequate to insure that software is developed and controlled according to standards.

Countermeasure. Establish software development procedures that place explicit controls on the software development process. These controls should cover the areas of program design, coding, and documentation.

a. Program design should include controls that cover the following:

(1) Audit trails to establish an historical record of processing.

(2) A thorough and comprehensive test plan covering program testing.

(3) Controls on the accuracy of data, such as input verification, matching against legal values, control fields, and self-checking digits.

(4) Quantitative controls, such as transaction counts, batch control totals, controls on rounding errors, reasonableness checks, and error suspense files.

b. Program coding should comply with such programming controls as the following:

(1) Organize programers in teams, make sure that no single programer is responsible for an entire sensitive application system.

(2) Observe naming conventions so that all references to a data element within an application are known by the same name.

(3) Use comments explaining accompanying code segments. These comments ease the task of program maintenance and help provide documentation.

(4) Use standardized indentation of source code to improve both readability and maintainability.

(5) Have a second programer inspect every program before it is compiled to make sure it conforms to standards, does not use restricted functions, and is logically complete.

c. Program documentation should be standardized within the ADP facility and thorough documentation should be required on all programs. Documentation should contain the following:

(1) A functional description of the program written in a narrative form describing the initial definition of the program and any subsequent changes.

(2) A program/subprogram section that contains information about the hardware environment, design elements, and interfaces.

(3) A program specification section that describes the program inputs, outputs, functions performed, interdependencies, and exception conditions.

(4) A program manual section with flowcharts, source listings, cross-reference listings, test data used, and operating instructions.

These standards may have to be adapted to individual facility needs.

Confidence. A medium level of confidence can be justified by this countermeasure.

Cost. The costs associated with this countermeasure will consist of the following:

a. The one-time administrative cost to establish procedures for this countermeasure is approximately $5,000 or one man-month.

b. The recurring administrative cost to review and update these procedures periodically is approximately $1,250 or one man-week per year.

c. The cost of recurring personnel time to comply with these procedures is estimated at five to ten percent of coding time for documentation writing and 10 to 15 percent of coding time for checking by a second qualified programer.

Caveats. Standards and conventions can be difficult to enforce and can add to initial software costs but may ease program maintenance.

1.4.2 Software Maintenance Procedures.

Vulnerability. The procedures governing the maintenance of production computer software may have weaknesses that lead to a compromise of security.

Countermeasure. Establish software maintenance procedures that place explicit controls on the software maintenance process. Controls on the software maintenance procedures should include the following:

a. Approved "Request for Change" should be required to initiate changes in production programs.

b. Program changes should be coded, tested, and documented in accordance with the facility software development and software acceptance procedures.

These controls may have to be adapted to individual facility needs.

Confidence. A medium level of confidence can be justified by this countermeasure.

Cost. The costs associated with this countermeasure will consist of the following:

a. The one-time administrative cost to establish procedures for this countermeasure is approximately $2,500 or two man-weeks.

b. The recurring administrative cost to review and update these procedures periodically is approximately $1,250 or one man-week per year.

c. The cost of recurring personnel time to comply with these procedures is estimated at 30 percent of the coding time.

Caveats. Standards and conventions can be difficult to enforce. These software maintenance procedures add to the initial cost of software maintenance but reduce the number of reruns that otherwise would occur because of maintenance errors in software modification.

1.4.3 Input/Output Procedures.

Vulnerability. An ADP facility may have inadequate procedures for the acceptance and release of information.

Countermeasure. Establish input/output procedures that place explicit controls on the submission of input and receipt of output. The input/output procedures should include the following:

a. Require users to submit job requests to use an ADP facility, such as to enter data or to make a production run.

b. Identify persons authorized to submit and pick up work from the ADP facility.

c. Control housekeeping activities to maintain the flow of work through the ADP facility.

d. Give all users instructions for obtaining and returning tapes and disks to the magnetic media library.

e. Provide instructions for the quality control of output and determination of correct security classifications.

f. Provide instructions to cover the signing of receipts upon receiving classified material and obtaining a receipt for classified output.

Confidence. A medium to high level of confidence can be justified by this countermeasure.

Costs. The costs associated with this countermeasure will consist of the following:

a. The one-time administrative cost to establish procedures for this countermeasure is approximately $2,500 or two man-weeks.

b. The recurring administrative costs to review and update these procedures periodically is approximately $1,250 or one man-week per year.

c. The continuing salary costs of the persons appointed to the input/output control group.

1.4.4 Access Procedures.

Vulnerability. Inadequate procedures for controlling access to supplies, computer equipment, and facilities can lead to unauthorized disclosure, theft, fraud, etc.

Countermeasure. Establish procedures for controlling access to the ADP facility, supply storage area, and other associated sites such as remote terminal areas and backup sites.

a. Procedures for controlling access to the ADP facility include the following:

(1) Access lists.
(2) Escort procedures.
(3) Identification badges.
(4) Guards.
(5) Mechanical or electronic door locks.
(6) Prompt removal of transferred or terminated employees from access lists and the mandatory turn-in of any facility identification or access keys or cards.

b. Periodic inventories should be conducted of computer equipment and related supplies.

Confidence.

a. The level of confidence placed in controlling access to the ADP system should be medium to high.

b. The level of confidence in procedures for accounting of computer equipment and related supplies should be medium to high.

Cost. The costs associated with this countermeasure will consist of the following:

a. Costs for procedures controlling access to the ADP facility will be:

(1) The one-time administrative cost to establish these procedures is approximately $1,250 or one man-week.

(2) The recurring administrative cost to review and update these procedures is approximately $2,500 or two man-weeks per year.

(3) The one-time cost to install whatever access control method is selected (see countermeasure 1.7.1, "Access to the Computer Center").

(4) The ongoing cost of maintaining whatever method of access control is selected.

b. The costs of procedures for accounting for computer equipment and related supplies will be recurring administrative costs to conduct periodic inventories.

1.4.5 Waste Procedures.

Vulnerability. Procedures may be inadequate to dispose of ADP waste materials.

Countermeasure. Establish procedures that clearly define the ADP waste materials that are to be disposed of in a secure manner and provide the facilities for secure disposal. These procedures should identify and provide destruction facilities for the following:

a. Paper and paper products, including carbon paper.

b. Printer ribbons.

c. Magnetic tapes, disks, drums, memory, etc.

d. Microfilm and microfiche if used.

Destruction facilities include incinerators, shredders, disintegrators, pulp machines, magnets, and tape degaussers.

Confidence. The level of confidence in this countermeasure is very high.

Cost. The costs associated with this countermeasure will be:

a. The one-time administrative cost to develop these procedures, estimated to be $250 or one man-day.

b. The one-time cost to purchase and install whatever method of destruction is selected. Shredders and disintegrators range in price from $2,000 to $6,000.

Caveats. Method of destruction for ADP waste materials must meet DOD standards.

1.4.6 Emergency Procedures.

Vulnerability. Security procedures for emergency situations may be inadequate, absent, or unenforceable.

Countermeasure. Establish well-conceived and technically feasible emergency procedures and test these procedures periodically. Sources of advice for the development of these procedures are the following:

a. The installation fire marshall.

b. The facility engineer.

c. The installation security office.

These procedures will normally cover the following:

a. Provide for off-site storage of duplicate records and files.

b. Arrange for processing critical applications at other ADP facilities.

c. Identify material to be evacuated or destroyed.

d. Designate a single point of contact for developing emergency procedures.

e. Provide transportion in the case of emergency evacuation.

Confidence. The level of confidence in this countermeasure should be medium to high.

Cost. The costs of this countermeasure will consist of the following:

a. The one-time administrative cost to establish these procedures is approximately $10,000 or two man-months.

b. The recurring administrative cost to review and update these procedures periodically should be approximately $2,500 or two man-weeks per year.

c. An undetermined one-time cost to provide the facilities to carry out these procedures.

d. An undetermined recurring cost to exercise these procedures periodically.

Caveats. Training and periodic exercises are essential to insure that emergency procedures would be carried out in an actual emergency.

1.4.7 Operations Procedures.

Vulnerability. The operations procedures may be inadequate and lead to disclosure, destruction, or a denial of service.

Countermeasure. Establish operations procedures that clearly and explicitly state how the ADP facility will function on a day-to-day basis. Some of the points that these procedures should cover are the following:

a. System start-up, shutdown and system crashes.
b. Priority scheduling of production runs.
c. Computer operations personnel interface with users and programers.
d. Separation of duties.
e. Rotation of duties.

Confidence. The level of confidence in this countermeasure should be medium to high.

Cost. The costs of this countermeasure will consist of the following:

a. The one-time administrative cost to establish these procedures is approximately $5,000 or one man-week of effort.

b. The recurring administrative cost to review and update these procedures is approximately $2,500 or two man-weeks per year.

c. The cost of training exercises on a periodic basis is calculated on the frequency, duration, and number of personnel involved.

1.4.8 Security Procedures and Security Officer.

Vulnerability. Security is a full-time job and each ADP facility should have an ADP System Security Officer (ADPSSO). The ADPSSO must have adequate authority to conduct an appropriate security program.

Countermeasure. Establish the position or function of ADPSSO and appoint someone in writing to fill the position or carry out the function. The ADPSSO should be located within the ADP facility organizational structure so that the

ADPSSO reports directly to the ADP facility commander or manager. Some of the functions of the ADPSSO should be:

a. Serves as the single point of contact for ADP security at the ADP facility.

b. Analyzes the ADP environment to identify vulnerabilities, assess threats, and apply countermeasures when needed.

c. Develops, maintains, and documents security requirements and operating procedures.

d. Insures that all personnel who install, operate, maintain, or use the ADP system know system security requirements and their responsibilities.

e. Establishes methods for detecting, reporting, investigating, and resolving ADP security incidents.

f. Establishes procedures for controlling changes to system hardware, software, applications, passwords, and central facility and terminal access.

g. Conducts periodic audits of security procedures and controls.

Confidence. The level of confidence in this countermeasure should range from medium to very high.

Cost. The costs associated with this countermeasure will consist of:

a. The recurring cost of the full or part-time salary of the individual appointed as ADPSSO.

b. The one-time cost of training the individual appointed as ADPSSO.

1.5 PERSONNEL

1.5.1 Personnel Controls.

Vulnerability. Poor management attitude and policy can lead to lapses in security.

Countermeasure. To prevent lapses in security, management should actively comply with security regulations and control procedures and make sure that employees do the same. Management should also seek out ways to improve security. Training and indoctrination courses should be given regularly to employees.

Confidence. Medium.

Cost. The cost depends on the time needed to develop a training course and the man-hours required to train each employee.

Countermeasure. To prevent employee misuse of or damage to the ADP facility, screen potential employees for personal integrity, stability, and conscientiousness. Maintain close and effective communications with the staff to prevent employee dissatisfaction or to deal with complaints if they arise.

Confidence. Medium.

Cost. The cost depends on the man-hours expended on pre-hire screening and management participation.

Countermeasure. To improve safety and security, periodically observe the work environment and work habits of employees. Observation will detect poor housekeeping habits that may increase the possibility of physical losses, such as tapes left on heaters, trash left in computer room, or coffee cups on equipment. Observation will also detect poor work habits that may compromise security, such as listings left unattended or files left open for unauthorized browsing.

Confidence. Medium.

Cost. Minimal.

Vulnerability. The personnel of the ADP system or facility can represent a degree of vulnerabiity that could be exploited to compromise security.

Countermeasure. To reduce the vulnerability of a compromise of classified defense information, require all personnel with unescorted access to the ADP facility to have a security clearance. The level of the security clearance must be at least as high as the level of information being processed. Uncleared personnel must be escorted by authorized persons and sensitive information must be protected.

Confidence. High.

Cost. The approximate cost to issue a TOP SECRET security clearance by the Defense Industrial Security Clearance Office is between $1,000 and $10,000.

Countermeasure. To reduce the risk of inadvertent damage by personnel, employ competent and well-trained personnel. Make clear the duties and obligations of employees.

Confidence. Medium.

Cost. None.

1.6 EMANATIONS

1.6.1 Emanations Security.

Vulnerability. Some components of a computer and computer peripherals emanate data signals various distances when processing or displaying data. These emanated data signals can be recorded by hostile monitoring equipment.

Countermeasure. Measures to control compromising emanations (TEMPEST) are required on systems that process classified information under the provisions of DOD Directive S-5200.19, 10 February 1968, "Control of Compromising Emanations (U)." There are basically three methods recommended for controlling

compromising emanations:

a. To provide the equipment with a physical control zone[2] sufficient to preclude successful hostile intercept actions.

b. To implement minimum essential countermeasures to contain compromising signals within a physical control zone.

c. To design or modify the equipment to limit the strength of possible compromising signals to acceptable limits considering the physical control zone available.

The equipment must be tested to determine the physical control zone required, the countermeasures which are required, or the possible equipment modifications needed. A physical control zone does not necessarily require a fenced area, guarded area, or a closed-circuit surveillance system, provided sufficient control is maintained to prohibit access for an unauthorized effort. (See Military Standardization Handbook 232[U].) The physical control zone requirement may be satisfied by security measures currently in place.

Confidence. If a large enough physical control zone is available or the minimum essential countermeasures are implemented or the equipment is designed or modified to limit the strength of emanations, then a high degree of confidence is gained.

Cost. The costs of containing compromising emanations vary depending on the equipment to be purchased and installation costs.

1.6.2 Radios and Tape Players.

---

2 Physical control zone is defined as the space surrounding equipment which processes classified information, that is under sufficient physical and technical control to preclude a successful hostile intercept of classified information from within this space.

Vulnerability. Radios, tape players, and other personally-owned equipment may be transmitters of electromagnetic emanations, which in turn may be modulated by nearby ADP equipment.

Countermeasure. Radios, tape players, and personally-owned electronic equipment should be banned from the computer room at installations processing classified information. Exceptions to this ban include equipment that has been technically inspected and approved.

Confidence. The strict enforcement of this countermeasure insures a high degree of confidence.

Cost. The costs of this countemeasure are the following:

a. The initial adminstrative cost of one man-week ($1,250) to develop the procedure for this countermeasure.

b. The recurring cost to conduct another technical inspection each time the equipment is brought back into the secure area.

1.7 PHYSICAL

1.7.1 Access to the Computer Center.

Vulnerability. The physical aspects of the ADP facility may make it difficult to control access to the facility.

Countermeasure. To prevent intruders from gaining access to the installation install an external surveillance system. Elements of the system include an external lighting system, a roving guard patrol, and closed circuit television surveillance. Also install intruder alarms on all unattended windows and doors.

Confidence. High.

Cost. The cost of intruder alarm systems and television surveillance is highly dependent on building design, sophistication of equipment to be used, and local labor rates.

Countermeasure. To prevent unauthorized persons from entering the installation or ADP facility, establish a guard force. The guard will verify and admit authorized personnel, maintain a visitor log, and insure that visitors are properly escorted.

Confidence. Medium.

Cost. The cost of guard service depends on local labor rates.

Countermeasure. To prevent unauthorized persons from entering the ADP facility, install an access system. The following systems provide protection by requiring the entrant to unlock a door. These systems may be used singly or in combination.

a. Conventional key and lock set
b. Electronic key system
c. Mechanical combination locks
d. Electronic combination locks

Confidence. High.

Cost. The following list provides the approximate cost per door:

a. Conventional key and lock set - $15
b. Electronic key system - $400 or more
c. Mechanical combination locks - $40 or more
d. Electronic combination locks - $500 or more

Caveats. The environment control system and storage rooms should also be secured to prevent unauthorized access.

Vulnerability. The physical layout inside the ADP facility may make it difficult to control the movement of persons within the ADP facility.

Countermeasure. To prevent unauthorized access to the computer room or other critical areas, such as tape library or communication equipment area, install an access system. The same types of locked-door systems described in the previous countermeasure are applicable.

Confidence. High.

Cost. See previous countermeasure.

Countermeasure. To prevent unauthorized access to critical areas that are unattended, install a passive security system. The following detection devices can be used singly or in combination:

a. Photometric system
b. Motion detection system
c. Acoustical system
d. Proximity system

Confidence. Very high.

Cost. The following list provides the approximate cost for each system, not including installation costs:

a. Photometric system - $500
b. Motion detection system - $250 or more
c. Acoustical system - $100
d. Proximity system - $350

Countermeasure. To minimize access to the computer area, access should be on a need-to-know basis. Visitors, maintenance personnel, and customer engineers should provide positive identification and always be escorted.

Confidence. Medium.

Cost. Minimal.

1.7.2 Fire Protection.

Vulnerability. The fire protection may be inadequate, making the ADP system or facility vulnerable to fire.

Countermeasure. Install a fire detection system. Place additional fire detectors above false ceilings, below raised floors, and in air conditioning ducts. Install a control panel that can identify the location of the detector that causes an alarm.

Confidence. High.

Cost. Approximately $3,500 plus installation costs for a 2,000 square foot room.

Countermeasure. Make fire extinguishers available in accessible locations. Mark each extinguisher as to the type of fire for which it is to be used. For example, a class *A* extinguisher should only be used on paper or wood.

Confidence. Medium.

Cost. A class *A* extinguisher costs approximately $35. A class *BC* extinguisher costs approximately $60.

Countermeasure. To provide a means of extinguishing or controlling a fire in the ADP facility, install an automatic fire extinguishing system. Three types of systems are: a water sprinkler system, a carbon dioxide system, and a HALON-1301 deluxe system. Install alarms to alert personnel if the system is activated. A water flow alarm can be used for sprinkler systems and a pressure sensor alarm can be used for gaseous systems.

Confidence. Very high.

Cost. The approximate cost for each system follows:

a. Water sprinkler system in a new building is $1.00 per square foot.

b. Retrofitted water sprinkler system is $3.00 per square foot.

c. HALON-1302 system is $.30 per cubic foot plus installation.

d. Carbon dioxide system is $.42 per cubic foot plus installation.

Countermeasure. Provide a fire protection plan to prevent the cause of fire and to extinguish a fire quickly. Develop the fire plan with the aid of the fire marshall. Conduct frequent inspections to identify and eliminate potential fire hazards.

Confidence. Medium.

Cost. Minimal.

Countermeasure. To protect equipment when a fire is detected, install and clearly mark emergency power disconnect switches. Make plastic sheeting available to cover equipment to protect against water damage. Store magnetic tapes and removable disk packs in fireproof or fire-resistant containers or rooms.

Confidence. Medium.

Cost. A fireproof safe that can store 48 magnetic tape reels costs approximately $2,000.

1.7.3 Environmental Control Systems.

Vulnerability. The environmental support systems (air conditioning, heating, and humidity control) may be inadequate to meet the mission's performance requirements.

Countermeasure. To protect against the failure of the air handling unit (AHU), install multiple units. For example, use three 20-ton AHUs in place of one 50-ton unit. There should be enough capacity to maintain the environment with one unit out of service. The air handling units circulate the computer room air, provide temperature and humidity control, and filter the air.

Confidence. High.

Cost. The approximate cost of an AHU is $350 per ton plus installation costs.

Countermeasure. To protect against the failure of the heating or cooling unit (compressor, heat pump, or circulation pump), install multiple units. There should be adequate capacity with one unit out of service.

Confidence. High.

Cost. The approximate cost of a cooling unit is $800 per ton plus installation costs.

Countermeasure. If the environmental control system fails, the capability to use outside air may be beneficial. Depending on location and weather, the use of direct outside air via vents and fans may be sufficient to maintain the temperature and humidity of the facility.

Confidence. Low.

Cost. The cost depends on the extent of modificatons and on local labor rates.

Countermeasure. Install an AHU designed to use and recirculate inside air in the event that outside air becomes unusable. The outside air may contain noxious fumes or may be of such poor quality that the filtration system would not be useful.

Confidence. Medium.

Cost. The cost depends on the extent of modifications and on local labor rates.

1.7.4 Building Construction.

Vulnerability. The construction of the building for the ADP system may introduce vulnerability.

Countermeasure. To protect against water damage caused by flooding, install pumps to remove water. Make sure that floor drains contain check valves to prevent water from entering the computer room. Install curbs around the facility or seal walls to divert water and prevent seepage.

Confidence. High.

Cost. A pump may cost from $600 to $5,000, depending on the size required.

Countermeasure. To prevent accidental flooding from plumbing failure, re-route pipes from above the facility. If this cannot be done, make sure that shutoff valves are accessible and clearly identified. Water pipes can be instrumented to detect any abrupt loss of pressure and to alert personnel.

Confidence. High.

Cost. The cost depends on the extent of modifications and on the local labor rate.

Countermeasure. To protect against damage caused by an earthquake, locate the ADP facility in a building with high resistance to earthquake damage. The building should be located to minimize risk of damage from neighboring buildings or structures.

Confidence. High.

Cost. Unknown.

Countermeasure. To protect against a fire outside the ADP facility, install fire walls and fire doors. Install fire dampers in all ducts leading to the facility to prevent smoke from entering.

Confidence. High.

Cost. The approximate cost of a fire door is $170 plus installation.

1.8 COMMUNICATIONS

1.8.1 Communications Lines and Links.

Vulnerability. It is possible to tap or monitor surreptitiously a data communication line or link; any data passed along the communications lines or links are susceptible to hostile interception or manipulation.

Countermeasure. Transmission and communication lines and links between components of an ADP system must be secured at a level appropriate for the material to be transmitted.

In the case of classified material, the countermeasures for secure communications lines or links are mandated by DOD Directives. Contact Naval Electronics Systems Security Engineering Facility.

For sensitive information or Privacy Act data, secure transmission is not mandated. However, during transmission some security should be provided, especially for sensitive data such as procurement data. This type of protection could be achieved through the use of the National Bureau of Standards Data Encryption Standard (DES), published as Federal Information Processing Standard (FIPS) Publication Number 46.

Confidence. A very high level of confidence can be placed in the mandated cryptographic techniques used for classified information. A high level of confidence can be placed in the DES.

Cost. A hardware implementation of the DES should be less than $5,000.

Caveats. An important factor affecting the level of confidence in the DES is the type of security afforded the keys used by the DES. A secure method must be developed to distribute the keys to the users of the system. Use of the DES in software is not approved at this time.

1.8.2 Terminal Identification.

Vulnerability. Many systems have improper or insufficient authentication of hardware. This can lead to a situation where an operating system cannot properly identify a terminal before responding to a request for data from the terminal. There is the possibility that data will be routed to a terminal whose location is not secure enough to support the storage of the data.

Countermeasure. Each remote terminal should be individually identified by a hardware feature in synch with the operating system. That is, the communications port, channel, and subchannel number should always communicate with the same remote terminal unless physically switched at the central site.

Confidence. A high level of confidence can be assigned to this countermeasure.

Cost. The costs associated with this countermeasure can be substantial if new terminals must be bought that have a unique identification symbol associated with them.

Caveats. Confidence in this countermeasure depends on how often the hardware identification symbol in each terminal series is repeated. The manner of implementation of the hardware identification, i.e., whether an identification is secure, also affects the confidence level.

1.8.3 Terminal Identification by Call Back.

Vulnerability. Many systems have insufficient means to identify valid terminal users. Without some form of terminal identification there is no way to assure that a potential penetrator terminal is not asking to be connected.

Countermeasure. For systems that allow dial-up terminal connection and that do not have an automatic hardware terminal identification feature, install a call-back terminal identification procedure.

The call-back procedure identifies a terminal dialing into a computer system. The central computer first disconnects the calling terminal, then re-establishes the connection by dialing the telephone number of the calling terminal.

Confidence. A call-back terminal identification procedure assumes a high level of confidence that the terminal asking to be connected is at least in the same location as the authorized terminal.

Cost. The costs associated with this countermeasure should be less than $2,000. This cost may be reduced by having the computer operator or switchboard operator perform the call-back procedure rather than installing automatic call-back software.

Caveats. If a procedure is set up to have either the computer console operator or a switchboard operator perform the call-back procedure, they must be instructed to call back the number at the place where the terminal is supposed to be, rather than a number that the terminal operator gives them.

1.8.4 Handshaking.

Vulnerability. Without some type of authentication procedure, there is no way a system and a user can identify each other.

Countermeasure. Handshaking is a procedure by which a system and a user (also two users or two systems) exchange identifiers to verify each other's identity. The identifiers can be passwords or even the successful execution of an algorithm.

Confidence. The confidence gained by using this countermeasure is medium.

Cost. The costs associated with this countermeasure are in software development and they should be no greater than one man-month ($5,000).

Caveats. Confidence in this countermeasure can vary as the protocol for handshaking varies. An exchange of common passwords, such as name of person and idenfication number of a hardware unit, will lower the level of confidence. Using a pseudo-random number transformation may raise the level of confidence.

1.8.5 Telephone Instruments.

Vulnerability. Factory-installed microphones in the handset and ringers on standard telephone instruments can act as microphones, either through design or intentional manipulation. They can then be used to monitor surreptitiously data signals and voices in the area around the telephone instruments.

Countermeasure. Relocate telephone instruments 1.8 meters (6 feet) or more from equipment used to process classified information. To minimize the technical security hazard posed by factory-supplied telephone instruments, remove the factory-supplied ringer from each instrument and install a protective ringing device.

Confidence. You can be assured of a high level of confidence that this countermeasure will substantially lower the risk that classified emanations or audio will travel out of a secure area over non-secure telephone instruments.

Cost. The cost per telephone will be less than $50.

1.8.6 Protected Wireline Distribution System (PWDS).

Vulnerability. Unprotected communications links can cause an ADP facility to be vulnerable to unauthorized activities including wiretapping and spoofing.[3]

Countermeasure. In some classes of ADP systems, it is desirable to employ a Protected Wireline Distribution System (PWDS). A PWDS is a telecommunications system that has been approved by a legally designated authority. It is a system for which electromagnetic and physical safeguards have been applied to permit safe electrical transmission of unencrypted sensitive information. Contact the ADPSSO for assistance.

Confidence. A PWDS will assure a high level of confidence that the communications system is secure against unauthorized activities.

---

[3] Spoofing: The deliberate inducement of a user or a resource to take an incorrect action.

Cost. The costs associated with a PWDS are high. However, the costs of a PWDS for a secure terminal area within the same building but outside the secure computer center may be less expensive than cryptographic protection.

Caveats. Once a PWDS has been run to a terminal or group of terminals, it is expensive to move the terminals and the PWDS lines.

1.8.7 Communications Path Alternatives.

Vulnerabilty. A communications system may be totally reliant on a set of communications paths. If one path becomes unavailable, serious denial of service problems may occur.

Countermeasure. Each path in a communications system should have an alternative route. There should be more than one way to get from one node to another node in the communications system.

Backup paths can be established physically (with hardware) and logically (with software).

Confidence. The greater the connectivity, i.e., the more alternative routes there are, the greater the confidence will be that a serious denial of service will not occur. One backup path gains a medium level of confidence at best.

Cost. It is expensive to retrofit software and hardware for establishing alternative paths in a communications system. Depending on the size of the communications system, this cost may be as high as $1 million. The cost of modifications to the software only is still high because of the complexity of this software.

Caveats. Some communications manufacturers have been successful in solving this problem. The G.E. Mark III network claims more than 99 percent uptime.

2.1 REFERENCES

2.1.1 References for 1.2.1

a. DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

b. L. J. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.

c. Federal Information Processing Standard (FIPS) Publication 41, May 1975.

d. J. Daly and G. Humphrey, Security Monitoring Near-Term Capability Status Report, TM-WD-5765/000/00, System Development Corporation, June 1975.

e. J. Daly and G. Humphrey, Security Monitoring Feasibility Analysis, TM-WD-5764/000/01, System Development Corporation, July 1975.

f. J. Daly, C. Dresser and G. Humphrey, Security Monitoring Concept Formulation, TM-WD-5770/001/10, System Development Corporation, June 1975.

g. R. G. McKenzie and Z. G. Ruthberg, "Computer Science and Technology: Audit and Evaluation of Computer Security," Proceedings of the National Bureau of Standards Invitational Workshop, Miami Beach, Fla., March 1977.

h. D. E. Denning, "A Method for Maintaining Routing Data in Automated Record Keeping Systems," Proceedings of Computer Software and Applications Conference, Institute of Electrical and Electronics Engineers, November 1978.

2.1.2 References for 1.2.2

a. D. Hsiao, D. Kerr, and S. Madnick, "Operating Security System, A Tutorial of Current Research," Proceedings of Computer Software and Applications Conference, Institute of Electrical and Electronics Engineers, November 1978.

b. K. S. Shankar, "The Total Computer Security Problem: An Overview," Computer, June 1977.

c. E. L. Burke, "Discovering Illicit Computer Usage," Proceedings COMPCON, Institute of Electrical and Electronics Engineers, International Conference, 1976.

d. C. Engleman, Audit and Surveillance of Multilevel Computing Systems, ESD-TR-76-369, The MITRE Corporation, April 1977.

2.1.3 References for 1.2.3

a. DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

b. R. R. Linde and R. F. Von Buelow, Jr., EXEC-8 Security Analysis, NRL Memorandum Report 3205, System Development Corporation, January 1976.

2.1.4 References for 1.2.4

a. World-Wide Military Command and Control System (WWMCCS) ADP System Security Officer (WASSO) Manual, SM-635-77, 25 July 1977.

b. L. J. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.

2.1.5 References for 1.2.5

a. M. Gasser, A Random Word Generator, MTR-3006, The MITRE Corporation, Bedford, Mass., 1975.

b. H. M. Wood, "On-Line Password Techniques," Institute of Electrical and Electronics Engineers Computer Society Symposium Proceedings, Trends and Applications 1977: Computer Security and Integrity, 1977.

2.1.6 Reference for 1.2.6

L. J. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.

2.1.7 References for 1.2.7

a. R. Bayer and J. K. Metzger, "On the Encipherment of Search Tree and Random Access Files," Transactions on Database Systems, Volume 1, Number 1, March 1976.

b. W. Diffie and M. Hellman, "Privacy and Authentication: An Introduction to Cryptography," Proceedings of the Institute of Electrical and Electronics Engineers, March 1979.

c. Data Encryption Standard, Federal Information Processing Standards ([FIPS] Publication 46), National Bureau of Standards, U.S. Department of Commerce, Washington, D.C., January 1977.

d. R. E. Lennon, "Putting Data Encryption to Work," Mini-Micro Systems, December 1978.

2.1.8 References 1.2.8

a. J. Kam and J. Ullman, A Model of Statistical Data Bases and Their Security, TR-207, Department of Electrical Engineering, Princeton University, 1976.

b. D. Denning, Are Statistical Data Bases Secure?, AFIPS National Computer Conference, 1978.

c. G. Davida, D. Wells, and J. Kam, "Security and Privacy," Proceedings of Computer Software and Applications Conference, Institute of Electrical and Electronics Engineers, 1978.

d. G. Davida, D. Linton, C. Szelag and D. Wells, "Data Base Security," Institute of Electrical and Electronics Engineers Transactions on Software Engineering, Volume SE-4, Number 6, November 1978.

2.1.9 References for 1.2.9

a. L. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.

b. B. Ruder and J. D. Madden, Computer Science and Technology: An Analysis of Computer Security Safeguards for Detecting and Preventing Intentional Computer Misuse, NBS 500-25, National Bureau of Standards, January 1978.

2.1.10 References for 1.2.12

a. World-Wide Military Command and Control System (WWMCCS) ADP System Security Officer (WASSO) Manual, SM-635-77, 25 July 1977.

b. DOD 5200.28-M, Techniques and Procedures for Implementing, Deactivating, Testing and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

c. DIAM 50-4, Security of Compartmented Computer Operations, January 1975.

d. J. M. Schacht, Jobstream Separator: Supportive Information, ESD-TR-75-354, The MITRE Corporation, January 1976.

2.1.11 Reference for 1.2.13

DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

2.1.12 References for 1.2.14

a. J. Martin, Security, Accuracy, and Privacy in Computer Systems, Prentice-Hall Inc., Englewood Cliffs, N. J., 1973.

b. J. Fitzgerald, Internal Controls for Computerized Systems, E. M. Underwood, Publisher, San Leandro, Calif., 1978.

2.1.13 References for 1.2.15

a. Handbook for Analyzing the Security of Operating Systems, Lawrence Livermore Laboratories, November 1976.

b. L. M. Culpepper and R. Regen, AUDIT: A System for Software Engineering for the CDC 6000, Naval Ship Research Development Center, November 1974.

c. "Automatic Generation of Self-Metric Software," Institute of Electrical and Electronics Engineers, Symposium on Computer Software Reliability, MDAC Paper WD2144, March 1973.

d. L. A. Clarke, "A System to Generate and Symbolically Execute Programs," Institute of Electrical and Electronics Engineers Transactions on Software Engineering, Volume SE-2, Number 3, September 1976.

e. Static FORTRAN Analyzer, Rome Air Development Center (ISIS), RADC-TR-75-275, November 1975.

f. D. Venese, Survey of Software Engineering Tools, TM-WD-7852/000/00, System Development Corporation, October 1977.

g. J. E. Sullivan, Measuring the Complexity of Computer Software, MTR-2648, Vol. V., The MITRE Corporation, June 1973.

h. T. J. McCabe, A Complexity Measure, Institute of Electrical and Electronics Engineers Transactions on Software Engineering, Vol. SE-2, Number 4, December 1976.

i. D. J. Reifer, Automated Aids for Reliable Software, SAMSO-TR-75-183, The Aerospace Corp., El Segundo, Calif., August 1975.

2.1.14 References for 1.2.16

a. J. P. Anderson, Computer Security Technology Planning Study, Volume II, Electronic Systems Division (MCIT), Air Force Systems Command, Bedford, Mass., ESD-TR-73-51, October 1972.

b. L. Galie, An Evaluation of Secure Subsystems for WWMCCS, TM-WD-5758/000/00, System Development Corporation, June 1975.

c. J. Gilligan, Secure Subsystem Development Study, TM-WD-7840/000/01, System Development Corporation, 15 April 1977.

d. J. Gilligan, S. DuGoff, and R. Stewart, Secure Transaction Processing Subsystem Concept of Operations, TM-WD-8007/000/01, System Development Corporation, March 1979.

e. J. A. Painter and J. S. Jackson, "Experience Using Secure Subsystems to Enhance Computer Security," Proceedings of the 1977 International Conference on Crime Countermeasures, Oxford, England, July 1977.

2.1.15 References for 1.2.17

a. Secure Minicomputer Operating System (KSOS), Computer Program Development Specifications (Type B-5), WDL-TR-7934, Ford Aerospace and Communications Corporation, September 1978.

b. E. J. McCauley, "KSOS: A Secure Operating System Study," COMPCON 79 Proceedings, February 1979.

c. J. P. Anderson, Computer Security Technology Planning Study, ESD-TR-73-51, Electronic Systems Division (MCIT), Air Force Systems Command; Bedford, Mass., October 1972.

d. M. Schaefer, On the Concept of the Security Kernel, TM-WD-5636/000/01, System Development Corporation, October 1976.

e. W. Schiller, Design of a Security Kernel for the PDP 11/45, MTR-2934, The MITRE Corporation, Bedford, Mass., March 1975.

f. C. H. Bonneau, Security Kernel Specifications for a Secure Communications Processor, ESD-TR-76-359, Honeywell, September 1976.

g. N. Adleman, J. R. Gilson, R. J. Sestak and R. J. Ziller, Security Kernel Evaluation for Multics and Secure Multics Design, Development and Certification, ESD-TR-76-298, Honeywell, August 1976.

2.1.16 References for 1.2.18

a. J. P. Buzen and V. O. Gagliardi, "The Evolution of Virtual Machine Architecture," Proceedings 1973, National Computer Conference, Volume 42, AFIPS Press, 1973.

b. C. R. Attanasio, Operating System Architecture and Integrity, IBM Data Security Forum, Denver, Colo., G520-2965-0, September 1974.

c. J. F. Sheid et al., Virtual Machine Monitor System Analysis, TM-WD-5443/001/01, System Development Corporation, September 1975.

d. L. J. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.

e. M. Shaefer et al., "VM/370 Security Retrofit Program," Proceedings, Association for Computer Machinery Annual Conference, October 1977.

f. M. Schaefer et al., "Program Confinement in KVM/370," Proceedings, Association for Computing Machinery Annual Conference, October 1977.

2.1.17 References for 1.2.19

a. A. Evans, W. Kantrowitz, and E. Weiss, "A User Authentication Scheme Not Requiring Secrecy in the Computer," Communications of the Association for Computer Machinery, Volume 17, Number 8, August 1974.

b. B. Purdy, "A High Security Log-In Procedure," Communications of the Association for Computer Machinery, Volume 17, Number 8, August 1974.

2.1.18 References for 1.3.1

a. DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

b. A. A. Bushkin, A Framework for Computer Security, TM-WD5733/000/00, System Development Corporation, June 1973.

c. B. J. Walker and I. F. Blake, Computer Security and Protection Structures, Dowden, Hutchinson & Ross Inc., Stroudsburg, Penn., 1977.

2.1.19 References for 1.3.2

a. DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

b. B. J. Walker and I. F. Blake, Computer Security and Protection Structures, Dowden, Hutchinson & Ross, Inc., Stroudsburg, Penn., 1977.

c. L. J. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.

2.1.20 Reference for 1.3.3

World-Wide-Military Command and Control System (WWMCCS) ADP System Security Officer (WASSO) Manual, SM-635-77, 25 July 1977.

2.1.21 References for 1.3.4

a. T. Hinke, D. Kaufman, and R. Mandel, Functional Specifications for Security Controls Within an ADP Computer Network, TM-WD-7831/001/00, System Development Corporation, 17 March 1977.

b. D. Venese and C. Gwinn, Network Security Analysis, TM-WD-7833/000/02, System Development Corporation, 23 March 1977.

c. G. D. Cole, Security/NFE Integration Plan, TM-WD-7831/002/01, System Development Corporation, 17 March 1977.

d. M. A. Branstad and D. K. Branstad, "Computer Security Application Utilizing Minis," National Bureau of Standards, Proceedings of Fall COMPCON, Institute of Electrical and Electronics Engineers, 1974.

2.1.22 References for 1.3.5

a. G. Cady and L. Wallace, A Preliminary Analysis of the DBM Concept for WWMCCS ADP, TM-WD-7803/000/01, System Development Corporation, August 1976.

b. G. Cady, L. Sveinsson, and D. Widrig, WWMCCS Data Base Machine Concepts, Requirements, and Functional Approach, TM-WD-7841/000/01, System Development Corporation, May 1977.

c. R. Canabey, R. Harrison, E. Svie, J. Ryder, and L. Wehr, "A Back-End Computer for Data Base Management," Communications of the Association for Computing Machinery, October 1974.

d. J. Banerjee, R. S. Buam, and D. K. Hsiao, "Concepts and Capabilities of a Database Computer," Association for Computing Machinery Transactions on Database Systems, December 1978, Volume 3, Number 4.

e. H. S. Ames, "RDM-A Relational Database Machine," Lawrence Livermore Laboratories, February 1977.

2.1.23 Reference for 1.3.6

J. M. Carroll, Computer Security, Security World Publishing Co. Inc., Los Angeles, 1977.

2.1.24 References for 1.3.7

a. A. A. Bushkin, A Framework for Computer Security, TM-WD-5733/000/01, System Development Corporation, June 1975.

b. H. R. Burris, "Microprogrammed Tamper Detection for Network Processors: Preliminary Results, "Symposium Proceedings Trends and Applications 1977: Computer Security and Integrity," Institute of Electrical and Electronics Engineers, May 1977.

2.1.25 References for 1.3.8

a. Federal Information Processing Standard (FIPS) Publication 31, June 1974.

b. J. Fitzgerald, Internal Controls for Computerized Systems, E. M. Underwood, Publisher, San Leandro, Calif, 1978.

c. D. B. Hoyt, Computer Security Handbook, Macmillan Information, Publisher, New York, 1973.

d. Westinghouse Electric Corporation, Consultants Guide to Uninterruptable Power Supply Systems, Buffalo, N. Y., May 1978.

2.1.26 References for 1.4.1

a. O. J. Dahl, E. W. Dykstra, and C. A. R. Hoare, Structured Programming, Academic Press, London and New York, 1972.

b. J. M. Buxton, P. Naur, and B. Randell, editors, Software Engineering Concepts and Techniques, Proceedings of the NATO Conference, Mason/Charter Publishers, Inc., New York, 1976.

c. K.R. London, "Documentation," Encyclopedia of Computer Science, Mason/Charter Publishers, Inc., New York, 1976.

d. DOD Manual 4120.17-M, Automated Data Systems Documentation Standards Manual, December 1972.

e. National Bureau of Standards (NBS), Federal Information Processing Standards (FIPS) Publication 31, June 1974.

f. W. P. Stevens, G. J. Myers, and L. L. Constantine, "Structured Design," IBM Systems Journal, Volume 13, Number 2, pages 115-139, 1974.

g. H. Bratman and T. Court, "The Software Factory," Computer, May 1975.

h. F. Gey, Professional Levels of Computer Program Documentation, Lawrence Berkeley Laboratory, Berkeley, California, 1976.

2.1.27 References for 1.4.2

a. M. I. Sobol, "Control of Program Maintenance," EDPACS: The EDP Audit, Control, and Security Newsletter, Automation Training Center, Inc., Reston, Va., February 1979.

b. R. J. Beach, "Maintaining Software Written in a Higher Level Language," Utility Software Subcommittee HLSUA Forum XXII, Toronto, May 1976.

c. W. T. Porter, Jr., "A Control Framework for Electronic Systems," Computers, Auditing and Control, Auerbach Publishers Inc., Wellington, Surrey, England, 1973.

d. H. E. Dickson, "Software Controls and Security," Computer Security Handbook, Macmillan Information, New York, 1973.

e. P. H. Braverman, "Managing Change," Datamation, Volume 22, Number 10, October, 1976.

2.1.28 References for 1.4.3

a. B. Ruder and J. D. Madden, "An Analysis of Computer Security Safeguards for Detecting and Preventing Intentional Computer Misuse," NBS Special Publication 500-25, U.S. Department of Commerce, January 1978.

b. J. Martin, Security, Accuracy, and Privacy in Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1978.

c. J. Fitzgerald, Internal Controls for Computerized Systems, E. M. Underwood, Publisher, San Leandro, Calif., 1978.

d. National Bureau of Standards (NBS), Federal Information Processing Standards (FIPS) Publication 31, June 1974.

2.1.29 References for 1.4.5

a. DOD 5200.1-R.

b. DOD 5200.28-M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

2.1.30 References for 1.4.6

a. L. Lettieri, "Disaster Recovery: Picking up the Pieces," Computer Decisions, Volume 11, Number 3, March 1979.

b. J. Kistner et al, Air Force Data Services Center Interim Emergency Procedures for Central Completer Facility, ESD, MTR-2880, The MITRE Corporation, Bedford, Mass., June 1974.

2.1.31 References for 1.4.7

a. J. Fitzgerald, Internal Controls for Computerized Systems, E. M. Underwood, Publisher, San Leandro, Calif., 1978.

b. B. J. Walker and I. F. Blake, Computer Security and Protection Structures, Dowden, Hutchinson and Ross, Inc., Stroudsburg, Penn., 1977.

2.1.32 References for 1.4.8

a. DOD 5200.28.

b. Department of Defense Computer Institute, Computer System Security Design Workshop Course Notes, 22 July to 1 August 1975.

c. World-Wide Military Command and Control System (WWMCCS) ADP System Security Officer (WASSO) Manual, SM-635-77, 27 July 1977.

2.1.33 References for 1.5.1

a. DOD 5200.1-R.

b. DOD 5200.28.

c. DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resources Sharing in ADP Systems, January 1973.

d. Federal Information Processing Standard (FIPS) Publication 31, June 1974.

e. D. B. Hoyt, Computer Security Handbook, Macmillan Information, Publisher, New York, 1973.

2.1.34 References for 1.6.1

a. World-Wide Military Command and Control System (WWMCCS) ADP System Security Officer (WASSO) Manual, SM-635-77, 25 July 1977.

b. Military Standardization Handbook 232(U).

2.1.35 Reference for 1.6.2

John M. Carroll, Computer Security, Security Publishing Co. Inc., Los Angeles, Calif., 1977.

2.1.36 References for 1.7.1

a. Federal Information Processing Standard (FIPS) Publication 31, June 1974.

b. Federal Information Processing Standard (FIPS) Publication 41, May 1975.

c. D. B. Hoyt, Computer Security Handbook, Macmillan Information, Publisher, New York, 1973.

d. J. Martin, Security, Accuracy, and Privacy in Computer Systems, Prentice-Hall Inc., Englewood Cliffs, N. J., 1973.

e. B. J. Walker and I. F. Blake, Computer Security and Protection Structure, Dowden, Hutchinson & Ross Inc., Stroudsburg, Penn., 1977.

2.1.37 References for 1.7.2

a. National Fire Protection Standard 75, paragraphs 2303, 3302, 4306, 5100, 5103, 5104, 5500, 5202, and 7301.

b. Federal Information Processing Standard (FIPS) Publication 31, June 1974.

c. D. B. Hoyt, Computer Security Handbook, Macmillan Information, Publisher, New York, 1973.

2.1.38 Reference for 1.7.3

Federal Information Processing Standard (FIPS) Publication 31, June 1974.

2.1.39 References for 1.7.4

a. R. Wright, S. Kramer, and C. Culver, Building Practices for Disaster Mitigation, National Bureau of Standards, February 1973.

b. U.S. Department of the Army, Flood-Proofing Regulations, Office of the Chief of Engineers, June 1972.

c. Federal Information Processing Standard (FIPS) Publication 31, June 1974.

d. D. B. Hoyt, Computer Security Handbook, Macmillan Information, Publisher, New York, 1973.

2.1.40 References for 1.8.1

a. DOD 5200.1-R.

b. Federal Information Processing Standard (FIPS) Publication 46, January 1977.

c. R. C. Lennon, "Cryptography Architecture for Information Security," IBM Systems Journal, Volume 17, Number 2, 1978.

d. S. M. Matyas and C. R. Meyer, "Generation, Distribution, and Installation of Cryptographic Keys," IBM Systems Journal, Volume 17, Number 2, 1978.

e. W. F. Ehisam, S. M. Matyas, C. H. Meyer, and W. L. Tuckman, "A Cryptographic Key Management Scheme for Implementing the Data Encryption Standard," IBM Systems Journal, Volume 17, Number 2, 1978.

2.1.41 Reference for 1.8.2

DOD 5200.28M, Techniques and Procedures for Implementing, Deactivating, Testing, and Evaluating Secure Resource Sharing in ADP Systems, January 1973.

2.1.42 References for 1.8.4

a. H. M. Wood, "The Use of Passwords for Controlling Access to Remote Computer Systems and Services," AFIPS Conference Proceedings, National Computer Conference, June 1977.

b. L. J. Hoffman, Modern Methods for Computer Security and Privacy, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1977.

2.1.43 Reference for 1.8.5

Military Standardization Handbook 232.

2.1.44 References for 1.8.6

a. World-Wide Military Command and Control System (WWMCCS) ADP System Security Officer (WASSO) Manual, SM-635-77, July 1977.

b. Military Standardization Handbook 232.

2.1.45 References for 1.8.7

a. K. S. Shankar and G.S. Chandersekaran, "The Impact of Security on Network Requirements," Symposium Proceedings Trends and Applications 1977: Computer Security and Integrity, Institute of Electrical and Electronics Engineers, May 1977.

b. B. Ruder and J. D. Madden, Computer Science and Technology: An Analysis of Computer Security Safeguards for Detecting and Preventing Intentional Computer Misuse, Special Publication 500-25, National Bureau of Standards, January 1978.